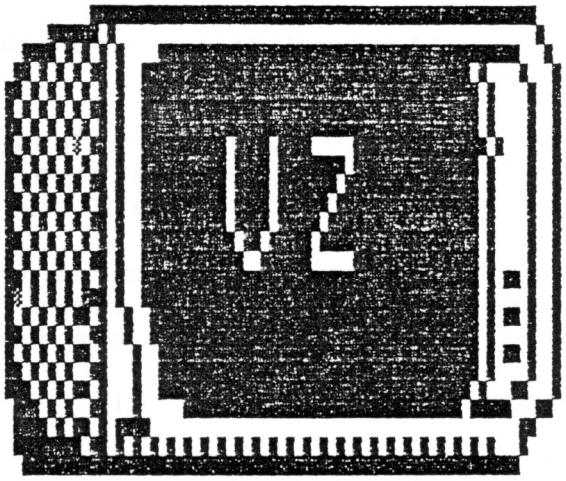
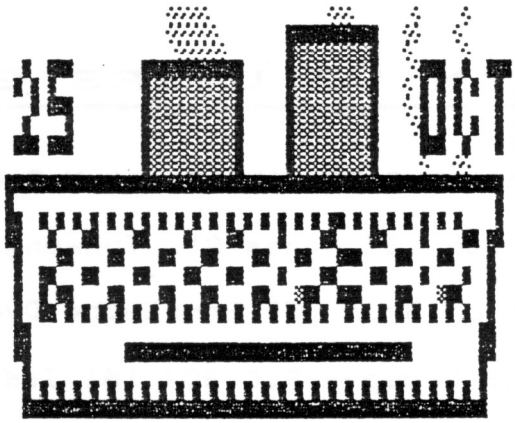
VZ-LINK MAGAZINE



ISSUE 25 OCTOBER 88



NEW ZEALAND

VZ 300/200

COMPUTER USERS CLUB

NEWS AND VIEWS

As you can see I have made a few changes to the newsletter format. I hope they meet with your approval.
The cover is drawn using a very fine program " ART GALLERY " written by bruce kitch.
The meeting/workshop was really well attended.I wish to thank all those that made the effort.

Thank's also to those that brought their equipment along for us to see.
It was good to meet Whittaker Hamilton who came all the way from taupo and Martin Deynzer from warkworth.

I would like to say hello to Richard Spong.
Poor richard not only broke his leg but also broke his arm later.
We'll hold some top scores open for you Richard.Get well soon mate.

Don't forget our competition,get your entry's in soon.
This issue is packed with programs and items for you all to enjoy.
If there is anything you would like to see in the newsletter,just let me know.
Also if you need help just give me a call.

See you next issue,cheers.....

AUSTRALIAN VZ USERS GROUPS

VZ USER---Mark Harwood P.O Box 154 Dural N.S.W 2158

LE'VZ OOP---John D'Alton 39 Agnes St. Toowong Qlnd 4066

VZ DOWN UNDER---Scott Le Brun 59 Brentwood Dve Wantirna 3152

HUNTER VALLEY VZ USERS Group---P.O BOX 161 Jesmond N.S.W 2299

WAVZ---Graeme Bywater P.O BOX 388 Morley W.A 6062

COMPETITION

All you have to do is design a LOGO for the VZ users club using 'high resolution' graphics on your VZ.
The LOGO must be suitable to use on the front of the newsletter and as a letter head.

The results will be judged on the way the program is written,as well as presentation.
Send your entry to me by november 19th.Judgeing will take place at the meeting on 26th November 88.

The prizes will be announced next issue.

ASSEMBLY LANGUAGE

Here's an example of how to POKE an assembly language program into memory from BASIC.

First of all write your assembly program.The program below fills the hi-res screen with a color.

```
001     LD  HL,7000H
002     LD  DE,7001H
003     LD  BC,07FFH
004     LD  (HL),0AAH
005     LDIR
006     RET
```

LD HL,7000H - Loads HL register pair with the HEX address of the first byte in video ram.7000 hex equals 28672 decimal.

LD DE,7001H - Loads DE register pair with the next byte along namely 7001 hex which equals 28673 decimal.

LD BC,07FFH - Loads BC with the size of video ram.7FF hex equals 2047 decimal.(2K of video ram)

LD (HL),0AAH - Loads 0AAH into memory location contained in the HL register pair.0AA hex equals 170 decimal.The 0 at the front of the hex number is to aid the DSE editor/assembler in recognising that it is a hex number.

LDIR - Repeat LDI until BC=0

Then enter it into your DSE Editor Assembler.Set the parameters by <SAC>.
This will list the assembly program along with assembled machine language code to your printer.Now <A>ssemble your program into machine code.Don't worry about where to <O>riginate the program,just use <00000>.

Now this is where the fun starts!!

```
001     LD  HL,7000H
0000 21 00 70
002     LD  DE,7001H
0003 11 01 70
003     LD  BC,07FFH
0006 01 FF 07
004     LD  (HL),0AAH
0009 36 AA
005     LDIR
000B ED B0
006     RET
000D C9
ERRORS : 00000
BYTES FREE :- 11501
```

Write down each of the hex numbers and convert these into decimal.

```
21 00  70 11 01 70 01 FF 07 36 AA  ED   B0  C9
33 00 112 17 1 112 01 255 7 54 170 237 176 201
```

Make sure that you don't get the order mixed up!

<<CONTINUED NEXT PAGE>>

Now all we need is somewhere to store our machine language program. I believe that the best place to put it is at the top end of memory.There is a pointer (memory location) which tells the computer where the TOP of MEMORY (TOM) is.So all we have to do is move this pointer down in memory by a number which is the length in bytes of your machine code program.

This will fool the computer into thinking that it has less memory available for BASIC programs than it actually has.The length of our program is 14 bytes.

To calculate your TOM.use :-PRINT PEEK(30897)+256*PEEK(30898) The number you should get is different(strangely enough) for different amounts of memory.Refer to table below.

36863-6K 45183-16K 53247-22K 63487-32k 65534-34K

Now you simply subtract the amount of space you want to reserve,namely 14 bytes.
So for example,say you had 63487 as your TOM.Your new TOM would be 63487-14=63473.
Now convert this number into hex giving F7F1.Pull this number apart into two bytes F7 and F1.Convert these back into decimal giving 247 and 241.

To use this,the first line of your BASIC program should be POKE 30897,241:POKE 30898,247.(always put the low byte first,then the high byte).

All you have to do now is to POKE your decimal numbers into the next memory location after your new TOM,which in this case would be 64474.
Go through the same steps and find the high and low order bytes of this new number which comes out as 247 and 242 respectively.These are the two numbers that you use to POKE into 30862/3 for your USR command to be activated.Here's how you should go about doing it.Type in the small program below.

```
10 POKE 30897,241:POKE 30898,247
20 FOR I=-2062 TO -2049
30 READ A:POKE I,A:NEXT
40 DATA 33,0,112,17,1,112,1,255,7,54, 170,237,176,201
50 CLS:PRINT"PRESS SPACE"
60 A$=INKEY$:B$=INKEY$
70 IF B$<>" ",60
80 MODE(1)
90  POKE30862,242:POKE30863,247
100 R=RND(255)
110 POKE-2052,R
120 X=USR(0)
130 FORI=1TO500:NEXT
140 GOTO 100
```

To explain the program. Line 10 creates the new TOM using the method described above.Lines 20-40 POKE in the machine code program. For some insane reason you cannot POKE over the address 32000.So the way to do it,is to subtract 65536 from your number giving you a negative number.

Lines 50-70 wait until <SPACE> is pressed.Line 90 sets up the USR command,by poking in the memory address to jump to when the USR command is encountered.
Lines 100-140 poke in a Random value into memory location -2052,which is the number being loaded into (HL) in our assembly language program.So by changing that number,you can change the pattern on the screen.Pretty nifty huh?.

WRITTEN BY SCOTT LE BRUN

Enhancing VZ Basic by Larry Taylor

The Commodore 64 has advanced hardware supported by an inadequate Basic language, resulting in a number of enhanced Basics being available. Something similar could be produced for the VZ. It must be noted, however, that all such Basics share a common disadvantage. Any program which makes use of them requires the language be loaded before it will function properly.

Because Basic is an interpreted language additional commands can be inserted, if they can be intercepted and executed before reaching the VZ's own interpreter. This is precisely what happens when a disk operating system (DOS) is added. New commands enabling disk operations to be performed, supplement the existing Basic. However, all programs using those extra commands require the DOS to be present before execution or they will not be interpreted correctly.

When a Basic program is RUN, control passes to a machine language ROM routine, the Execution Driver at 1D5AH, which scans each line of the Basic program as it comes to it and begins to translate it. Part of the translation process involves looking for tokens. These are values in the range 128-250 (80H-FAH) that take the place of Basic reserved words e.g. CLS = 132 (84H). Once the word has been identified and checked for correct syntax, control is passed to the corresponding ROM routine before returning to continue the translation. This is similar to one person issuing instructions to another through an interpreter, who first has to translate them before the receiver can act, and is the reason for Basic's slow execution. Most languages get around this problem by having the program translated or compiled before execution.

Tandy's Colour Computer has an enchanced CLS command which enables the user to clear the screen to any one of nine background colours. The syntax is CLSn, where n may be a number in the range 0-8. To illustrate how enhancements can be accomplished, this command will be added to the VZ's repertoire.

On power up the address of the routine which examines each byte in a line of Basic, is stored at 7804H. Because this address is in RAM it can be easily changed. This was done so that at a later stage the DOS could be included. However, it also means that, just as readily, an enhanced form of Basic may be added. The trick is to ensure that, as far as the VZ's interpreter is concerned, nothing unusual has happened. The accompanying assembly language listing shows how this can be accomplished.

ENHANCING VZ BASIC — L.TAYLOR

```
100 REM ##########################################################
105 REM # ENHANCED CLS COMMAND BY LARRY TAYLOR 1986              #
110 REM ##########################################################
115 REM # CALCULATE THE NEW TOP OF MEMORY POINTER                #
120 REM ##########################################################
125 NB=79:TM=(PEEK(30897)+PEEK(30898)*256)-NB
130 HB=INT(TM/256):LB=TM-HB*256
135 POKE30897,LB:POKE30898,HB
140 REM ##########################################################
145 REM # RESET THE BASIC STACK POINTER                          #
150 REM ##########################################################
155 CLEAR50
160 REM ##########################################################
165 REM # LOCATION OF SET UP PROGRAM                             #
170 REM ##########################################################
175 EB=31274
180 EH=INT((EB+1)/256):EL=EB+1-EH*256
185 REM ##########################################################
190 REM # LOAD USER EXECUTION PROGRAM POINTER                    #
195 REM ##########################################################
200 POKE30862,EL:POKE30863,EH
205 REM ##########################################################
210 REM # LOAD 22 BYTE SET UP PROGRAM                            #
215 REM ##########################################################
220 FOR T=1TO22
225  READD
230  POKEEB+T,D
235  CS=CS+D
240 NEXT
245 REM ##########################################################
250 REM # GET NEW TOP OF MEMORY AND MOVE TO NEXT LOCATION        #
255 REM ##########################################################
260 TM=PEEK(30897)+PEEK(30898)*256
265 IFTM>32767THENTM=TM-65536
270 REM ##########################################################
275 REM # LOAD 79 BYTE ENHANCED CLS ROUTINE                      #
280 REM ##########################################################
285 FOR T=1TO79
290  READD
295  POKETM+T,D
300  CS=CS+D
305 NEXT
310 REM ##########################################################
315 REM # IF DATA CHECKSUM VERIFIES EXECUTE SET UP PROGRAM       #
320 REM ##########################################################
325 IFCS<>10725THENPRINT"- ERROR IN DATA ENTRY -":END
330 X=USR(0)
335 REM ##########################################################
340 REM # SET UP EXECUTION ROUTINE DATA IN DECIMAL FORM          #
345 REM ##########################################################
350 DATA 49,0,119        :' LD SP,7700H
355 DATA 42,4,120        :' LD HL,(7804H)
360 DATA 34,40,122       :' LD (7A28H),HL
365 DATA 42,177,120      :' LD HL,(78B1H)
370 DATA 35              :' INC HL
375 DATA 34,4,120        :' LD (7804H),HL
380 DATA 205,77,27       :' CALL 1B4DH
385 DATA 195,25,26       :' JP 1A19H
```

```
390 REM ##########################################################
395 REM # ENHANCED CLS ROUTINE DATA IN DECIMAL FORM              #
400 REM ##########################################################
405 DATA 217              :' EXX
410 DATA 33,91,29         :' LD HL,1D5BH
415 DATA 209              :' POP DE
420 DATA 183              :' OR A
425 DATA 237,82           :' SBC HL,DE
430 DATA 213              :' PUSH DE
435 DATA 217              :' EXX
440 DATA 194,120,29       :' JP NZ,1D78H
445 DATA 229              :' PUSH HL
450 DATA 205,120,29       :' CALL 1D78H
455 DATA 32,7             :' JR NZ,07H
460 DATA 225              :' POP HL
465 DATA 237,91,40,122    :' LD DE,(7A28H)
470 DATA 213              :' PUSH DE
475 DATA 201              :' RET
480 DATA 254,132          :' CP 84H
485 DATA 32,245           :' JR NZ,F5H
490 DATA 35               :' INC HL
495 DATA 126              :' LD A,(HL)
500 DATA 214,48           :' SUB 30H
505 DATA 40,9             :' JR Z,09H
510 DATA 6,8              :' LD B,08H
515 DATA 184              :' CP B
520 DATA 40,4             :' JR Z,04H
525 DATA 16,251           :' DJNZ FAH
530 DATA 24,230           :' JR E6H
535 DATA 209              :' POP DE
540 DATA 209              :' POP DE
545 DATA 17,30,29         :' LD DE,1D1EH
550 DATA 213              :' PUSH DE
555 DATA 35               :' INC HL
560 DATA 229              :' PUSH HL
565 DATA 135              :' ADD A,A
570 DATA 135              :' ADD A,A
575 DATA 135              :' ADD A,A
580 DATA 135              :' ADD A,A
585 DATA 32,1             :' JR NZ,01H
590 DATA 60               :' INC A
595 DATA 198,127          :' ADD A,7FH
600 DATA 33,0,112         :' LD HL,7000H
605 DATA 34,32,120        :' LD (7820H),HL
610 DATA 17,1,112         :' LD DE,7001H
615 DATA 1,255,1          :' LD BC,01FFH
620 DATA 119              :' LD (HL),A
625 DATA 237,176          :' LDIR
630 DATA 225              :' POP HL
635 DATA 201              :' RET
```

MATH'S PROGRAM
BY WHITTAKER HAMILTON

```
38 CLS:PRINT:PRINT:PRINT:PRINT:PRINT
40 FOR J=1TO500:NEXT:PRINT"        PLEASE STOP THE TAPE"
41 SOUND 21,7 :FOR I=1TO500:NEXT
42 REM MATHEMATICS PROGRAM
43 REM ********************
44 REM * MADE BY WHITTAKER*
45 REM * HAMILTON FOR THE *
46 REM * VZ-300. (C) 1988.*
47 REM ********************
48 CLS:PRINT "   WELCOME TO THE DICK SMITH "
49 PRINT"     MATHEMATICS PROGRAM."
50 FORI=1TO2000:NEXT
51 CLS
52 PRINT"   PROGRAMMED AND DESIGNED BY"
53 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT"      WHITTAKER HAMILTON"
54 FORI=1TO2000:NEXT:CLS
58 CLS:PRINT"PLEASE ENTER YOUR NAME."
59 INPUT B$:CLS:PRINTB$
60 PRINT"DO YOU WANT TO BE TESTED ON     ";

65 PRINT"EASY OR HARD MULTIPLICATION"
70 INPUT CA$
80 IF CA$="HARD"THEN GOTO 250
85 IF CA$="E"THEN GOTO110
90 IF CA$="EASY"THEN GOTO 110
95 IF CA$="H"THEN GOTO250
100 GOTO 70
110 Z=0
115 GOSUB410
120 FOR I=1TO10
140 LET A=RND(10)
150 LET B=RND(10)
155 PRINT"QUESTION ";I
160 PRINT"WHAT DOES ";A;"TIMES ";B;"EQUAL"
165 PRINT B$
170 INPUT C
190 IF C=A*B THEN LET Z=Z+1
192 GOSUB 410
200 IF C=A*B THEN PRINT"WELL DONE ";B$:SOUND31,6
210 IF C<>A*B THEN PRINT"WRONG ";B$;" THE ANSWER WAS"
213 IF C<>A*B THEN PRINT A*B
214 IFC<>A*BTHENSOUND1,8
216 FORJ=1TO 1000:NEXT
218 GOSUB410
220 IF C<>A*BTHENPRINT"BETTER LUCK NEXT TIME ";B$
225 IF C<>A*B THEN GOSUB 1000
228 IFC<>A*BTHENGOSUB410
230 NEXT I
240 GOTO 500
250 Z=0
255 GOSUB410
260 FOR H=1TO10
280 LET A=RND(12)
290 LET B=RND(12)
300 PRINT"QUESTION ";H
```

```
310 PRINT"WHAT DOES ";A;" TIMES ";B;" EQUAL  ";B$
320 INPUT C
340 IF C=A*BTHENLETZ=Z+1
345 GOSUB 410
350 IF C=A*B THEN PRINT"WELL DONE ";B$:SOUND31,6
360 IF C<>A*BTHENPRINT"WRONG ";B$;" THE ANSWER WAS "
362 IF C<>A*BTHENPRINTA*B
363 IFC<>A*BTHENSOUND1,8
365 FOR X=1TO1000:NEXT
375 GOSUB 410
380 IF C<>A*BTHEN PRINT "BETTER LUCK NEXT TIME ";B$
385 FORW=1TO1000:NEXT
388 IFC<>A*BTHENGOSUB410
390 NEXT H
400 GOTO 500
410 CLS:PRINT"     MULTIPLICATION PROGRAM"
415 PRINT"     ************** *******"
420 PRINT
425 PRINT "DESIGNED AND PROGRAMMED FOR THE VZ-300 BY"
430 PRINT"    <<<WHITTAKER HAMILTON.>>>"
440 PRINT"    VZ-300 MICROSOFT (C)1988."
450 PRINT"================================"
460 RETURN
500 GOSUB410
510 PRINT"YOUR SCORE ";B$;" WAS ";Z;"/10"
515 FOR W=1TO1000:NEXT
525 GOSUB 410
530 PRINT Z;"/10"
540 PRINT"DO YOU WANT ANOTHER GO ";B$
550 INPUT A$
555 Q=2
560 IF A$="Y"THEN GOTO 58
570 IF A$="YES"THEN GOTO 58
580 IF A$="NO"THEN LET Q=1
590 IF A$="N"THEN LET Q=1
595 FOR P=1TO1000:NEXT
597 CLS
600 IF Q=1 THEN PRINT"DOES A FRIEND ";B$
610 INPUT F$
620 IF F$="YES"THEN GOTO 48
630 IF F$="Y"THEN GOTO 48
640 END
1000 FORS=1TO1000:NEXT
1010 RETURN
```

FOR SALE

ORIGINAL VZ PROGRAM TAPES
AND SMALL RANGE P.C.B,S

REASONABLE RANGE OF VZ TAPES AT AUST.$5.00 EACH
SMALL RANGE OF P.C.B'S ALSO AVAILABLE (P.O.A)

SEND SELF ADDRESSED/STAMPED ENVELOPE FOR
A LIST OF TAPES AND P.C.B'S TO:-

DAVE BOYCE
41 HEATHER DRIVE
CHRISTIE DOWNS 5146
SOUTH AUSTRALIA

PLEASE INCLUDE FULL NAME AND ADDRESS

Hardware Review

ETI-812

VZ Data Logger

This project is featured in the JULY 1988 edition of Eletronics Today International (ETI).

For $60 you can build a box to plug into your VZ200/300 computer to log up to 8 ***analogue*** channels. The data can then be stored on tape or with a simple program modification, saved on disk.

The eight input channels are configured as channels 0-6 for analogue data inputs and channel 7 used as a counter. Voltage range from +2.5V up to +3.56V. It has a sampling rate of 3 per second which is reasonable for most applications but is too slow for sampling speech or music.

The unit is powered directly from the VZ via the peripheral port, which means that you can't use either the printer or joysticks simulatenously with it. It uses the vacant port 64 to input all its information from the logger to the computer

Using 5 channels for 51 days continuous data sampling fills only 6K of RAM as 1 byte holds 1 data value.

The logger also has three digital outputs which could be used for indicators, alarms control, etc.

The article suggests using 2 pin din plugs, but I've found these to be unreliable when you solder them. I think RCA sockets would be of better value, and are cheaper, more readily available and are more widespread in use.

Bob Sutton, who designed the project, has written the software to be deliberatly simple, so that you can alter it to suit your own requirements. Bob has even listed some pokes so that you can save your data as a basic prgram (t:file).

There has been a flood of excellent VZ peripherals hitting the electronics magazines lately with this project being no exception.

REVIEW WRITTEN BY SCOTT LE BRUN

DISKLIST UPDATE BY JOE LEON

```
MAIN MENU          28-1-1988
RECORDS IN FILE XMASLIST= 24
==============================
1. ENTER DATA   0. CLEAR FILE
2. DOS DISK FUNCTIONS
3. VIEW/EDIT RECORDS
4. VIEW NAMES IN FILE
5. SORT INTO ALPHA. ORDER
6. ENTER TODAYS DATE
7. PRINT NAME & PHONE LIST
8. PRINT COMPLETE LIST
9. PRINT MAILING LABELS
==============================
WHICH FUNCTION (1-9)
```

```
DISK MENU          28-1-1988
==============================
R - REN  FILE   E - ERA  FILE
S - SAVE DATA   L - LOAD DATA
D - DIRECTORY   I - INIT DISK
M-MENU P-PEEK   V-SAVE T-REPL
==============================
MERGE DATA FILE - Y/N ? N
```

Some of the changes made to DISKLIST are cosmetic while others are more functional. As before attempt has been made to keep to original style and the extra functions easy to use.

POKE30777,1 in lines 30, 600, 1000,7000 and 40010 are POKEs that speed up PRINTing to the screen.

POKE30777,35 in lines 645 and 1205 turn speed printing off and it must be done before disk access to stop interference with drive.

Lines 49, 2010, 2020, 3250 and 3260 add COMMAND line to ENTER DATA and VIEW NAMES screens. Lines 2020 and 3260 only need PRINT@0, changed to PRINT@32, no need to retype whole lines.

LINES 850 to 915, 5330 to 5350 and 6110 are the modified disk SAVE/LOAD routines providing two extra functions. Lines 850 to 915 are presented whole as some routines have been shifted around and line numbers have been renumbered with some lines not being altered exept for line numbers

S - SAVE DATA - When you press 'S' an UPDATE DATA FILE Y/N option will be presented with the cursor flashing over the 'N'. Pressing RETURN will take you to normal SAVE. Entering 'Y' and pressing RETURN will allow you after entering filename to first ERAse your DATA FILE and then SAVE your DATA FILE using the same filename. In other words, automatic update.

L - LOAD DATA - When you press 'L' a MERGE DATA FILE - Y/N option will be presented with the cursor flashing over the 'N'. Pressing RETURN will take you to normal LOAD. Entering 'Y' and pressing RETURN will allow you to MERGE DATA FILES from disk with FILES in memory.

Only a few minor changes were needed to allow file merging. They are in lines 856, 868 and 6110.

0. CLEAR FILE - A clear file option has been provided. Previously this option was available only after SAVING a file, or you had to use BREAK and RUN program again. As the CLEAR FILE was already part of MAILING LIST it was easy to implement. Lines 1030, 1225, and 5330 to 5350 contain the changes.

In the Main Menu the FILENAME is now displayed and is the last change done to DISKLIST. Exept for Disk functions, most modifications can also be incorporated in MAILING LIST.

DISKLIST UPDATE CONT. . . . .

```
30 POKE30777,1:FORSC=33TO449STEP32

49 OP$="  QUIT  SCROLL  DELETE   PRINT"

600 POKE30777,1:GOSUB30:COLOR7:PRINT@34,"| DISK MENU |    ";DT$

645 PRINT@354,"| SELECT FUNCTION |       ":SOUND30,1:POKE30777,35

850 PRINT@354,SP$:PRINT@380,"N"
852 PRINT@354,"| MERGE DATA FILE (Y/N) | ";:SOUND30,1:INPUTN$
854 N$=LEFT$(N$,1):IFN$<>"N"ANDN$<>"Y"THEN850
856 IFN$="N"THENOF=0ELSEOF=DT
858 IFN$="Y"THEN880ELSEIFN$="N"THEN860ELSE850
860 PRINT@354,SP$
862 PRINT@354,"| LOAD DATA FILE |  ";:SOUND30,1:INPUTN$:GOSUB775
864 PRINT@354,SP$:PRINT@450,SP$:SOUND30,1:AD=31603:GOSUB22
866 AD=31619:GOSUB22:AD=31646:GOSUB22:AD=31674:GOSUB22
868 GOSUB6000:GOSUB20:DT=OF+DT:GOTO1000
878 :
880 IFDT<>0THEN884ELSEPRINT@353,"    | NO DATA IN MEMORY |    "
882 SOUND5,7:GOTO645
884 IFDM$="L"THEN860
886 PRINT@354,SP$:PRINT@380,"N"
888 PRINT@354,"| UPDATE DATA FILE (Y/N) | ";:SOUND30,1:INPUTN$
890 N$=LEFT$(N$,1):IFN$<>"N"ANDN$<>"Y"THEN886ELSEPRINT@354,SP$
892 IFN$="N"THENUP=0ELSEUP=1
894 :
896 PRINT@354,"| SAVE DATA FILE |  ";:SOUND30,1:INPUTN$:GOSUB775
898 PRINT@450,SP$;:SOUND30,1:IFUP=0THEN906
900 :
902 PRINT@354,SP$:PRINT@354,"| ERASING FILE |  ";N$:SOUND30,1
904 AD=31481:GOSUB22:GOSUB12
906 PRINT@354,SP$:AD=31533:GOSUB22:AD=31549:GOSUB22
908 AD=31576:GOSUB22:AD=31674:GOSUB22:GOTO5000
915 :

1000 POKE30777,1:IFDT$=""THENDT$="NO DATE"

1020 PRINT@66,"RECORDS IN FILE          ";:PRINT@82,N$
1025 PRINT@90,USING"=###";DT
1030 PRINT@98,MI$:PRINT@130,"1. ENTER DATA  0. CLEAR FILE";

1205 POKE30777,35

1225 IFI$="0"THENIFDT=0THEN1400ELSE5260

2010 CLS:PRINTOP$
2020 COLOR8:PRINT@32,"________________________________";

3250 CLS:PRINTOP$
3260 COLOR8:PRINT@32,"________________________________";

5330 IFI$="1"THEN5350ELSEIFI$="2"THEN5340ELSE5310
5340 DT=0:ST$="":N$="":DT$="NO DATE"
5350 IFDM$="S"THEN600ELSE1000

6110 FORN=OF+1TOOF+DT

7000 POKE30777,1:GOSUB40000:GOSUB30:COLOR7

40010 POKE30777,1:CLS:COLOR8,0
```

SOFTWARE FOR SALE

We now have the following software available for sale.
All prices shown include postage and packing anywhere in New Zealand.
Please make sure that the memory of your VZ is greater than or equal to the memory required by the software that you wish to purchase.
Memory sizes indicated do not include video ram.

ALL PRICES ARE SHOWN IN NEW ZEALAND DOLLARS.

BLACKJACK ROYALE	16K	$15.00	HAUNTED MANSION	22K	$15.00
CASTLE GREYSTONE	32K	$15.00	KNIGHTS QUEST	32K	$18.00
DATA	16K	$18.00	NAMESTAR/SHOPSTAR	22K	$15.00
FBI-2001	31K	$18.00	SCOTLAND YARD	16K	$ 8.00
GALACTIC EMPIRES	16K	$15.00	TRIVIAL CULT	16K	$ 8.00
GRAPHSTAR	22K	$24.00	VZ MONOPOLY V2.0	22K	$18.00

PLEASE MAKE ALL CHEQUES AND MONEY ORDERS PAYABLE TO-

AUCKLAND VZ 300/200 COMPUTER CLUB.

VZ DISASSEMBLER FOR SALE

This Disassembler is written by Peter Hickman.
And is also distributed as the "Laserlink Disassembler".

Peter is offering this very fine program to user group members for NZ$24.00.
All orders can be placed through this club,as peter has offered a cheaper price for bulk purchases.

The program is available on tape or disk.And comes in a 4 program version and
You choose which section you use depending on what memory you have.

ALL THESE PROGRAMS CAN BE DEMONSTRATED AT THE WORKSHOP IF REQUESTED

P.O.BOX 1972 C.P.O AUCKLAND N.Z

(C) AUCKLAND VZ 300/200 COMPUTER USER'S CLUB 1986

ALL MATERIAL IN THIS NEWSLETTER IS SUBJECT TO COPYRIGHT..
CONTRIBUTED PROGRAMS & ARTICLES CAN BE REPRODUCED ON THE UNDERSTANDING THAT THEY ARE FOR THE PRIVATE USE OF SUBSCRIBERS ONLY.
COPYRIGHT IS ALWAY'S RETAINED BY THE AUTHOR.
THIS CLUB WILL NOT BE HELD RESPONSIBLE FOR THE BREECHING OF COPYRIGHT CAUSED BY THE PUBLICATION OF ANY MATERIAL SENT IN UNDER A FALSE NAME.

YOU ARE WELCOME TO PHONE ME BETWEEN 6.30 PM AND 9 PM ----PHONE 673-967